

AWS DOCUMENTATION

Host the App

1)Launch Instance with Setting Security Group

Step I : Launch the Instance with adding two rules in Security group .Add Custom TCP : 8080 and MYSQL/Aurora :3306.

The screenshot displays the AWS Management Console interface for configuring Security Group rules. It shows two rules, 'Security group rule 2' and 'Security group rule 3', each with a 'Remove' button. Rule 2 is configured with Type 'Custom TCP', Protocol 'TCP', Port range '8080', Source type 'Anywhere', Source '0.0.0.0/0', and Description 'e.g. SSH for admin desktop'. Rule 3 is configured with Type 'MYSQL/Aurora', Protocol 'TCP', Port range '3306', Source type 'Anywhere', Source 'Add CIDR, prefix list or secur', and Description 'e.g. SSH for admin desktop'.

Rule ID	Type	Protocol	Port range	Source type	Source	Description
Security group rule 2 (TCP, 8080, 0.0.0.0/0)	Custom TCP	TCP	8080	Anywhere	0.0.0.0/0	e.g. SSH for admin desktop
Security group rule 3 (TCP, 3306, 0.0.0.0/0)	MYSQL/Aurora	TCP	3306	Anywhere	Add CIDR, prefix list or secur	e.g. SSH for admin desktop

Step II : Then Launch Instance and connect it.

2)Set up the Apache Tomcat and Deploy Application.

Step III : Download Apache Tomcat using “curl -O <https://dlcdn.apache.org/tomcat/tomcat9/v9.0.87/bin/apache-tomcat-9.0.87.zip>” this command.

```
[ec2-user@ip-172-31-25-9 ~]$ curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.87/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 11.7M  100 11.7M    0     0  7292k      0  0:00:01  0:00:01 --:--:-- 7291k
[ec2-user@ip-172-31-25-9 ~]$ ls
apache-tomcat-9.0.87.zip
[ec2-user@ip-172-31-25-9 ~]$ unzip ^C
[ec2-user@ip-172-31-25-9 ~]$ unzip apache-tomcat-9.0.87.zip
```

Step IV : then, unzip it.

Step V : Download Student Application file using “curl -O https://db-tom.s3.ap-northeast-1.amazonaws.com/student.war” this command.

Step VI : and then move this file in webapps directory using “mv student.war apache-tomcat-9.0.87/webapps/” this command.

```
apache-tomcat-9.0.87  apache-tomcat-9.0.87.zip
[ec2-user@ip-172-31-25-9 ~]$ curl -O https://db-tom.s3.ap-northeast-1.amazonaws.com/student.war
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 89423  100 89423    0     0  119k      0  --:--:-- --:--:-- --:--:-- 119k
[ec2-user@ip-172-31-25-9 ~]$ ls
apache-tomcat-9.0.87  apache-tomcat-9.0.87.zip  student.war
[ec2-user@ip-172-31-25-9 ~]$ mv student.war apache-tomcat-9.0.87/webapps/
```

Step VII : We need Java for Tomcat to work,so install java using “yum install java -y” command.

```
[ec2-user@ip-172-31-25-9 ~]$ sudo yum install java -y
Last metadata expiration check: 0:20:29 ago on Mon Mar 18 04:51:59 2024.
Dependencies resolved.
=====
Package                                Architecture      Version
=====
Installing:
```

Step VIII : then,we need to give permission to catalina.sh .

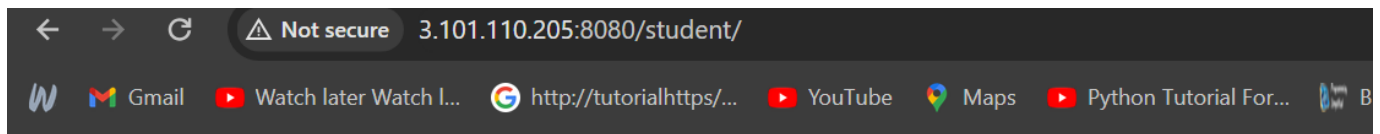
Step IX : for this “ `chmod 777 apache-tomcat-9.0.87/bin/catalina.sh`”using these command to give the execute permissions.

Step X : After giving the permissions ,Start the catalina using “`./apache-tomcat-9.0.87/bin/catalina.sh start`” these command.

```
Complete!
[ec2-user@ip-172-31-25-9 ~]$ chmod 777 apache-tomcat-9.0.87/bin/catalina.sh
[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh start
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/
Using CATALINA_OPTS:
Tomcat started.
```

Step XI : then go to instances, copy the public ip of these Instance and paste it in other browser.

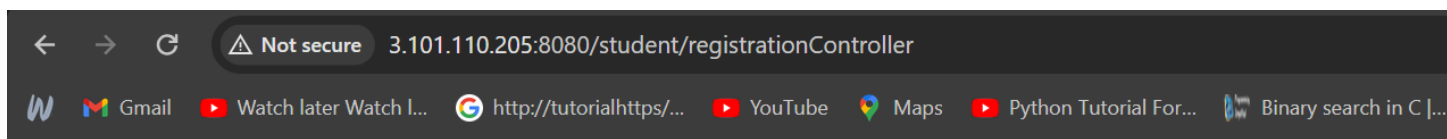
Step XII : and after IP add “:8080/student”.



Student Registration Form

Student Name	<input type="text" value="Apurva"/>
Student Address	<input type="text" value="Pune"/>
Student Age	<input type="text" value="21"/>
Student Qualification	<input type="text" value="MCS"/>
Student Percentage	<input type="text" value="89"/>
Year Passed	<input type="text" value="2025"/>
<input type="button" value="register"/>	

Step XIII : See here, after registering the data it is unable to save.



Sorry! unable to save record

3)Set up the database

Step I : to save the data permanently we need to create database.

Step II : go to the RDS service and click on databases.


Step III : then, create database .


Step IV : Select the Standard create,Engine type is MYSQL, select the free tier template .


CloudWatch


Engine options


Engine type [Info](#)


☐ Aurora (MySQL Compatible)


☐ Aurora (PostgreSQL Compatible)


☒ MySQL


☐ MariaDB


☐ PostgreSQL


☐ Oracle


Templates

Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**
This instance is intended for development use outside of a production environment.

☒ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Step V : then,enter the name for DB instance, username as a admin and give the Password for database.

▼ Credentials Settings

Master username

Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

☐ Managed in AWS Secrets Manager - *most secure*

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ Self managed

Create your own password or have RDS create a password that you manage.

☐ Auto generate password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password

Info

.....

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password

Info

.....|


Storage

Storage type [Info](#)

Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)

Baseline performance determined by volume size

 CloudWatch

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

▼ Hide filters

☒ Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

☐ Include previous generation classes

☐ Standard classes (includes m classes)

☐ Memory optimized classes (includes r and x classes)

☒ Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps



Step VI : then ,uncheck the storage auto scaling option.

▼ Storage autoscaling

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application

☐ Enable storage autoscaling

Enabling this feature will allow the storage to increase after the specified threshold

Step VII : then choose security group which we used in instance.

☒ No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing

Choose existing VPC security groups

☐ Create new

Create new VPC security group

Existing VPC security groups

Choose one or more options ▼

launch-wizard-5 ✕

Step VIII : then uncheck the Enable automated backups and encryption, auto minor version upgrade and click on create database.

Backup

- ☐ **Enable automated backups**
Creates a point-in-time snapshot of your database

Encryption

- ☐ **Enable encryption**
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Maintenance

Auto minor version upgrade [Info](#)

- ☐ **Enable auto minor version upgrade**
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

- ☐ Choose a window
☒ No preference

Deletion protection

- ☐ **Enable deletion protection**
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.



You are responsible for ensuring that you have all of the necessary rights and permissions that you use with AWS services.

Step IX : see here, database is generated.

4)Configure Tomcat to connect to RDS .

Step I : for connecting we need to MYSQL connector to connect database.

Step II : for downloading MYSQL connector give these command “curl -O <https://db-tom.s3.ap-northeast-1.amazonaws.com/mysql-connector.jar>”.

Step III : Move these file in the lib directory in apache – tomcat, using these “mv mysql-connector.jar apache-tomcat-9.0.87/lib/” command.

```
[ec2-user@ip-172-31-25-9 ~]$ curl -O https://db-tom.s3.ap-northeast-1.amazonaws.com/mysql-connector.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 983k  100 983k    0     0  920k      0  0:00:01  0:00:01 --:--:--  920k
[ec2-user@ip-172-31-25-9 ~]$ ls
apache-tomcat-9.0.87  apache-tomcat-9.0.87.zip  mysql-connector.jar
[ec2-user@ip-172-31-25-9 ~]$ mv mysql-connector.jar apache-tomcat-9.0.87/lib
[ec2-user@ip-172-31-25-9 ~]$ ls apache-tomcat-9.0.87/lib/
annotations-api.jar      catalina.jar             jsp-api.jar              tomcat-i18n-cs.jar      tomcat-i18n-es.jar
catalina-ant.jar         ecj-4.20.jar             mysql-connector.jar       tomcat-i18n-de.jar      tomcat-i18n-es.jar
catalina-ha.jar          el-api.jar               servlet-api.jar           tomcat-i18n-es.jar      tomcat-i18n-fr.jar
catalina-ssi.jar         jasper-el.jar            tomcat-api.jar            tomcat-i18n-fr.jar      tomcat-i18n-ja.jar
catalina-storeconfig.jar jasper.jar               tomcat-coyote.jar         tomcat-i18n-ja.jar      tomcat-i18n-ko.jar
catalina-tribes.jar      jaspic-api.jar           tomcat-dbcp.jar           tomcat-i18n-ko.jar      tomcat-i18n-ko.jar
[ec2-user@ip-172-31-25-9 ~]$ mv mysql-connector.jar apache-tomcat-9.0.87/lib/
```

Step IV : then edit the tomcat context.xml to add RDS data Configuration in it.

```
Resource name="jdbc/TestDB" auth="Container"
type="javax.sql.DataSource" maxTotal="100"
maxIdle="30" maxWaitMillis="10000"
```

username="admin" password="anvi1234"
driverClassName="com.mysql.jdbc.Driver"
url="ENDPOINT /DATABASE_NAME"/> • After making
the changes in context.xml save the file and exit.

```
<!-- The contents of this file will be loaded for each web application -->
<Context>
    <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="admin"
password="anvi1234" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://database-1.cjsuug66g5rg.us-west-1.rds.amazonaws.com/st

    <!-- Default set of monitored resources. If one of these changes, the
    <!-- web application will be reloaded.
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
```

Step V : then Install the mariadb using “yum install
mariadb105 -y”.

```
[ec2-user@ip-172-31-25-9 ~]$ sudo vim apache-tomcat-9.0
[ec2-user@ip-172-31-25-9 ~]$ sudo yum install mariadb105
```

5)Connect the Database

Step I : use the Command “mysql -h Endpoint -u
username -ppassword;”.

```
IAM CloudWatch
[ec2-user@ip-172-31-25-9 ~]$ mysql -h database-1.cjsuug66g5rg.us-west-1.rds.amazonaws.com -
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.35 Source distribution
```

Step II : once connected, then create a new database using “create database database_name;”.

Step III : Switch newly created database using “use db_name”.

Step IV : CREATE TABLE if not exists
students(student_id INT NOT NULL
AUTO_INCREMENT, student_name VARCHAR(100)
NOT NULL, student_addr VARCHAR(100) NOT NULL,
student_age VARCHAR(3) NOT NULL, student_qual
VARCHAR(20) NOT NULL, student_percent
VARCHAR(10) NOT NULL, student_year_passed
VARCHAR(10) NOT NULL, PRIMARY KEY (student_id)
);

Add this script in it use exit command to exit form db.

```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> create database studentapp;
Query OK, 1 row affected (0.003 sec)

MySQL [(none)]> use studentapp;
Database changed
MySQL [studentapp]> CREATE TABLE if not exists students(student_id INT NOT NULL
-> AUTO_INCREMENT,
-> student_name VARCHAR(100) NOT NULL,
-> student_addr VARCHAR(100) NOT NULL,
-> student_age VARCHAR(3) NOT NULL,
-> student_qual VARCHAR(20) NOT NULL,
-> student_percent VARCHAR(10) NOT NULL,
-> student_year_passed VARCHAR(10) NOT NULL,
-> PRIMARY KEY (student_id)
-> );
Query OK, 0 rows affected (0.047 sec)

MySQL [studentapp]> exit

```

Step V : then Stop the catalina.sh using “./apache-tomcat-9.0.87/bin/catalina.sh stop”.

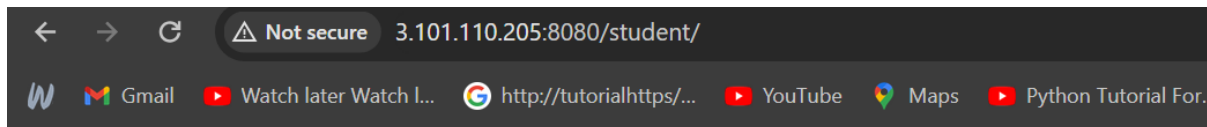
Step VI : add start again using “./apache-tomcat-9.0.87/bin/catalina.sh start”.

```

[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh stop
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/ap
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java
va.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rm
[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh start
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/ap
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-172-31-25-9 ~]$

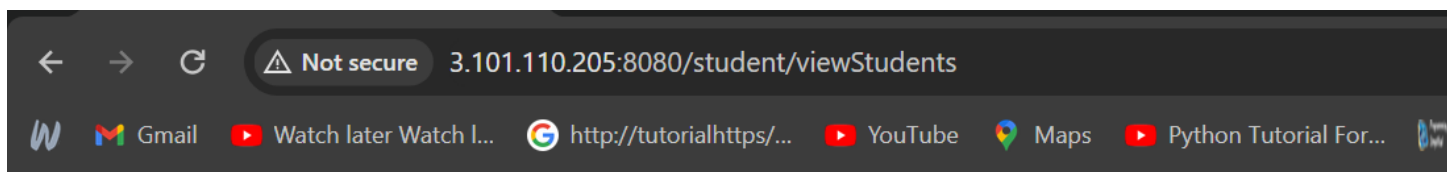
```

Step VII : see here data will be added successfully.



Student Registration Form

Student Name	<input type="text" value="anvi"/>
Student Address	<input type="text" value="pune"/>
Student Age	<input type="text" value="5"/>
Student Qualification	<input type="text" value="nur"/>
Student Percentage	<input type="text" value="99"/>
Year Passed	<input type="text" value="2024"/>
<input type="button" value="register"/>	



[Register Student](#)

Students List

Student ID	StudentName	Student Addr	Student Age	Student Qualification
1	anvi	pune	5	nur

Creating the Proxy server.

Step I : Launch the new Instance add same security group as we added in above instance and also add http.

Instances (1/2) Info							Connect	Instance state ▼	Actions
Find Instance by attribute or tag (case-sensitive)						Any state ▼			
	Name	Instance ID	Instance state	Instance type	S				
<input checked="" type="checkbox"/>	proxy	i-0766062dedcc7a36a	Running	t2.micro					
<input type="checkbox"/>	tomcat	i-08a88016d4aaca890	Running	t2.micro					

Step II : Connect the instance .

Step III : then Install nginx.

```

IAM CloudWatch
[ec2-user@ip-172-31-4-177 ~]$ sudo yum install nginx
Last metadata expiration check: 0:04:33 ago on Sat Mar 1
Dependencies resolved.
=====

```

Step IV : using ls check the file in /etc/nginx.

Step V : then edit the configuration file type “vim /etc/nginx/nginx.conf”.

Step VI : In this, Add the application address in this file.

Location / {

Proxy_pass http: //ApplicationPublicIP:8080/student/;
}

```
location / {  
    proxy_pass http://3.101.110.205:8080/student/;  
}
```

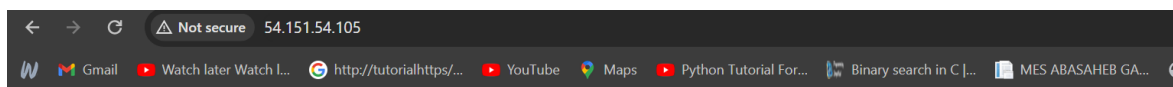
Step VII : then, Start the service using “systemctl start nginx” and enable it.

```
[ec2-user@ip-172-31-19-94 ~]$ sudo vim /etc/nginx/nginx.conf  
[ec2-user@ip-172-31-19-94 ~]$ sudo su  
[root@ip-172-31-19-94 ec2-user]# systemctl start nginx  
[root@ip-172-31-19-94 ec2-user]# systemctl enable nginx  
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service  
[root@ip-172-31-19-94 ec2-user]#
```

To Verify

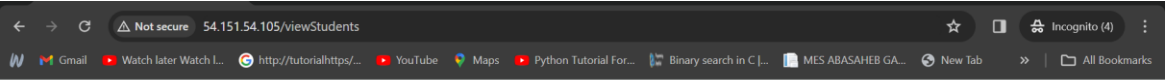
Copy the public IP of proxy server instance and paste it in another browser.

After registering student see here it is added successfully.



Student Registration Form

Student Name	<input type="text"/>
Student Address	<input type="text"/>
Student Age	<input type="text"/>
Student Qualification	<input type="text"/>
Student Percentage	<input type="text"/>
Year Passed	<input type="text"/>
<input type="button" value="register"/>	



[Register Student](#)

Students List

Student ID	StudentName	Student Addr	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	anvi	pune	5	nur	99	2024	edit	delete
2	apurva	mp	20	mcs	89	2025	edit	delete